09-18-00

A

PATENT

Attorney Docket No. MPATENT.167A
Date: September 14, 2000

ASSISTANT COMMISSIONER FOR PATENTS

WASHINGTON, D.C. 20231

ATTENTION: BOX PATENT APPLICATION

Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): **Duane Allen**

For: **BIOS LOCK ENCODE/DECODE DRIVER**

Enclosed are:

(X) 4 sheet(s) of drawing.

(X) Recordation form cover sheet with 1-page Assignment.

(X) Establishment of Right of Assignee to Take Action and Revocation and Power of Attorney.

(X) Initial signed declaration by inventor(s).

(X) Return prepaid postcard.

CLAIMS AS FILED

FOR	NUMBER FILED	NUMBER EXTRA	RATE	FEE
Basic Fee			$690	$690
Total Claims	26 - 20 =	6 ×	$18	$108
Independent Claims	7 - 3 =	4 ×	$78	$312
If application contains any multiple dependent claims(s), then add			$260	$0
	TOTAL FILING FEE			$1,110

(X) A check in the amount of $1,110 to cover the filing fee is enclosed.

(X) A check in the amount of $40 to cover the assignment recording fee.

(X) The Commissioner is hereby authorized to charge any additional fees which may be required, now or in the future,or credit any overpayment to Account No. 11-1410. A duplicate copy of this sheet is enclosed.

(X) Please use Customer No. 20,995 for the correspondence address.

Richard E. Campbell
Registration No. 34,790
Attorney of Record

S:\DOCS\REC\REC-4383.DOC 091300

KNOBBE, MARTENS, OLSON & BEAR, LLP
620 NEWPORT CENTER DR 16TH FLOOR NEWPORT BEACH, CA 92660
(949) 760-0404 FAX (949) 760-9502

INTELLECTUAL PROPERTY LAW

KNOBBE, MARTENS, OLSON & BEAR

A LIMITED LIABILITY PARTNERSHIP INCLUDING
PROFESSIONAL CORPORATIONS

PATENT, TRADEMARK AND COPYRIGHT CAUSES

550 WEST C STREET

SUITE 1200

SAN DIEGO, CALIFORNIA 92101-3505

(619) 235-8550

FAX (619) 235-0176

INTERNET WWW.KMOB.COM

LOUIS J. KNOBBE*
DON W. MARTENS*
GORDON H. OLSON*
JAMES B. BEAR
DARRELL L. OLSON*
WILLIAM B. BUNKER
WILLIAM H. NIEMAN
ARTHUR S. ROSE
JAMES F. LESNIAK
NED A. ISRAELSEN
DREW S. HAMILTON
JERRY T. SEWELL
JOHN B. SGANGA, JR
EDWARD A. SCHLATTER
GERARD VON HOFFMANN
JOSEPH R. RE
CATHERINE J. HOLLAND
JOHN M. CARSON
KAREN VOGEL WEIL
ANDREW H. SIMPSON
JEFFREY L. VAN HOOSEAR
DANIEL E. ALTMAN
MARGUERITE L. GUNN
STEPHEN C. JENSEN
VITO A. CANUSO III
WILLIAM H. SHREVE
LYNDA J. ZADRA-SYMES†
STEVEN J. NATAUPSKY
PAUL A. STEWART
JOSEPH F. JENNINGS
CRAIG S. SUMMERS
ANNEMARIE KAISER
BRENTON R. BABCOCK

THOMAS F. SMEGAL, JR
MICHAEL H. TRENHOLM
DIANE M. REED
JONATHAN A. BARNEY
RONALD J. SCHOENBAUM
JOHN R. KING
FREDERICK S. BERRETTA
NANCY WAYS VENSKO
JOHN P. GIEZENTANNER
ADEEL S. AKHTAR
GINGER R. DREGER
THOMAS R. ARNO
DAVID N. WEISS
DANIEL HART, PH.D.
DOUGLAS G. MUEHLHAUSER
LORI LEE YAMATO
MICHAEL K. FRIEDLAND
STEPHEN M. LOBBIN
STACEY R. HALPERN
DALE C. HUNT, PH.D.
LEE W. HENDERSON, PH.D.
DEBORAH S. SHEPHERD
RICHARD E. CAMPBELL
MARK M. ABUMERI
JON W. GURKA
ERIC M. NELSON
MARK R. BENEDICT, PH.D.
PAUL N. CONOVER
ROBERT J. ROBY
SABING H. LEE
KAROLINE A. DELANEY
JOHN W. HOLCOMB
JAMES J. MULLEN, III, PH.D.

JOSEPH S. CIANFRANI
JOSEPH M. REISMAN, PH.D.
WILLIAM R. ZIMMERMAN
GLEN L. NUTTALL
ERIC S. FURMAN, PH.D.
TIRZAH ABE LOWE
GEOFFREY Y. IIDA
ALEXANDER S. FRANCO
SANJIVPAL S. GILL
SUSAN M. MOSS
JAMES W. HILL, M.D.
ROSE M. THIESSEN, PH.D.
MICHAEL L. FULLER
MICHAEL A. GUILIANA
MARK J. KERTZ
RABINDER N. NARULA
BRUCE S. ITCHKAWITZ, PH.D.
PETER M. MIDGLEY
THOMAS S. MCCLENAHAN
MICHAEL S. OKAMOTO
JOHN M. GROVER
MALLARY K. DE MERLIER
IRFAN A. LATEEF
AMY C. CHRISTENSEN
SHARON S. NG
MARK J. GALLAGHER, PH.D.
DAVID G. JANKOWSKI, PH.D.
BRIAN C. HORNE
PAYSON J. LEMEILLEUR
WILLIAM G. BERRY
DIANA W. PRINCE

OF COUNSEL

JERRY R. SEILER
PAUL C. STEINHARDT

JAPANESE PATENT ATTY
KATSUHIRO ARAI**

EUROPEAN PATENT ATTY
MARTIN HELLEBRANDT

KOREAN PATENT ATTY
MINCHEOL KIM

SCIENTISTS & ENGINEERS
(NON-LAWYERS)

RAIMOND J. SALENIEKS**
DANIEL E. JOHNSON, PH.D.**
JEFFERY KOEPKE, PH.D.**
KHURRAM RAHMAN, PH.D.
JENNIFER A. HAYNES, PH.D.
BRENDAN P. O'NEILL, PH.D.
THOMAS Y. NAGATA
LINDA H. LIU
YASHWANT VAISHNAV, PH.D.
MEGUMI TANAKA
CHE S. CHERESKIN, PH.D.**
ERIK W. ARCHBOLD
PHILIP C. HARTSTEIN
JULIE A. HOPPER
CHRIS S. CASTLE
JAMES W. AUSLEY
R. P. CARON, PH.D.
JENNIFER HAYES
KIRK E. PASTORIAN, PH.D.
CHARLES T. RIDGELY
KEITH R. MCCOLLUM
LANG J. MCHARDY

* A PROFESSIONAL CORPORATION
† ALSO BARRISTER AT LAW (U.K.)
** U.S. PATENT AGENT

Assistant Commissioner for Patents
Washington, D.C. 20231

EL587855636US

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

Attorney Docket No.	:	MPATENT.167A
Applicant(s)	:	Duane Allen
For	:	BIOS LOCK ENCODE/DECODE DRIVER
Attorney	:	Richard E. Campbell
"Express Mail" Mailing Label No.	:	EL587855636US
Date of Deposit	:	September 14, 2000

I hereby certify that the accompanying Transmittal in Duplicate; Specification in 18 pages; 4 sheets of drawings; **Signed** Declaration by Inventor in 1 pages; Recordation Form Cover Sheet and Assignment in 2 pages; Establishment of Right of Assignee to Take Action and Revocation and Power of Attorney by Assignee in 2 pages; Check for Filing Fees; and Return Prepaid Postcard are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and are addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Fernando Guilarte Jr

Fernando Guilarte, Jr.

S:\DOCS\REC\REC-4386.DOC
091300

BIOS LOCK ENCODE/DECODE DRIVER

Background of the Invention

Field of the Invention

This invention relates to the field of software security and further relates to systems and methods for restricting any unauthorized use of software.

Description of the Related Art

It has become commonplace to sell computers including personal computers (PCs) with software. It has also become commonplace for the software to be provided on removable media such as compact disks (CD-ROM, compact disk-read only memory), zip disks, or floppy disks. The purchaser of the PC then loads the software onto the PC by placing the removable media into an appropriate removable media reader in the PC. The removable media reader then reads the software and the software is transferred to the memory of the PC.

Consumer software is typically licensed for use on a single computer. However, it is difficult to control the use of the software once it is in the possession of the consumer. Therefore, there is a need for systems and methods which prevent the unauthorized access or use of software and data on removable media.

Summary of the Invention

The present invention provides systems and methods for preventing the unauthorized access to encoded content stored on removable media. Such content includes computer software such as operating systems and application programs.

One aspect of the invention includes a computer system for controlling the unauthorized use of software. The system includes a host processor and a removable media reading device coupled to the host processor. A non-volatile memory is coupled to the host processor and includes a predetermined location for storing a signature. A bootup program is stored in the non-volatile memory and is configured such that upon execution by the host processor, the computer system will only be configured to decode encoded media which is placed in the removable media device if the signature is located in the predetermined location.

Another aspect of the invention includes a method of reading encoded data from a removable media device in a computer system. The method includes initiating a request for data from the removable media reader and determining whether the system is authorized to decode the encoded data. If the system is authorized to decode the encoded data, the method then determines whether the first sector on the removable media is encoded. If the system is authorized to decode encoded data and the first sector on the removable media is encoded, the requested data is decoded.

In a further aspect of the invention, determining whether the system is authorized to decode encoded data includes verifying that a signature is stored in the system in a predetermined memory location.

In another aspect of the invention, a method of preventing unauthorized access to encoded contents stored on removable media by a computer system running an operating system is disclosed. The method includes running a memory-resident program with a lockable decoding function to control the interaction between the operating system and the removable media reader. The method includes scanning the computer system for a predetermined signature and unlocking the lockable decoding function if the signature is found. Removable medium is examined to determine whether it is encoded. If the removable medium is encoded, the lockable decoding function is enabled. Finally, data from the removable medium is decoded if the lockable decoding function is enabled.

In another aspect of the method, the memory resident program is inserted between the operating system and a device driver for the removable media reader.

In a further aspect of the method, the memory resident program adds at least one modular driver between the operating system and the removable media reader.

In a further aspect of the method, the predetermined signature is stored in a basic input output system circuitry of the computer system.

In another aspect of the invention includes a method of preventing unauthorized access to encoded contents stored on removable media by a computer system running an operating system and having a removable media reader. The method includes inserting a driver with a lockable decoding function between the operating system and the device driver for the removable media reader to control the transfer of the information between

the operating system and the removable media reader. Further, the method includes scanning the computer system for a predetermined signature and unlocking the lockable decoding function if the signature is found. The method also includes examining a removable medium to determine whether it is encoded and enabling said lockable decoding function if the removable medium is encoded. Finally, data is decoded from the removable medium if the lockable decoding function is enabled.

Another aspect of the invention relates to a method of preventing unauthorized access to encoding content stored on removable media. The method includes adding one modular driver between an operating system and the removable media reader to incorporate a lockable decoding function to control the transfer of information between the operating system and the removable media reader. The method also includes scanning the computer system for a predetermined key and unlocking the lockable decoding function if the signature is found. Next, the method includes examining a removable medium to determine whether it is encoded and enabling the lockable decoding function if the removable medium is encoded. Finally, data from the removable medium is decoded if the lockable decoding function is enabled.

These and other aspects and advantages of the present invention will be come more apparent upon a detailed review of the following detailed description and the accompanying figures.

Brief Description of the Drawings

The features and advantages of the present invention will become more apparent from the following detailed description of the present invention in conjunction with the drawings in which:

Figure 1 is a block diagram of functional elements of a computer system, such as a personal computer;

Figure 2 is a flowchart representing the steps or process of bootup in a DOS system according to aspects of the present invention;

Figure 2a is a block diagram representing software drivers;

Figure 3 is a flowchart of the process or method which can be carried out when removable media is first inserted into a reader in a DOS-based system;

Figure 4 is a flowchart which represents the process or method of reading removable media according to aspects of the present invention;

Figure 5 is a flowchart which represents the process or method which occurs upon the loading of a NT/2000-type operating system according to aspects of the present invention;

Figure 6 is a flowchart which represents the process or method for determining whether removable media is encoded according to aspects of the present invention on a NT/2000-type operating system;

Figure 7 is a flowchart which represents the process or method for reading removable media according to aspects of the present invention on a NT/2000-type operating system;

Figure 8 is a flowchart which represents a process or method of encode detection carried out by a removable media reader; and

Figure 9 is a flowchart which represents a process or method of reading and decoding carried out by a removable media reader.

Detailed Description of the Preferred Embodiment

Figure 1 is a block diagram of an exemplary computer system 10. The computer system can include a chip set 51 which operates as an interface to support communications between a host processor 50, system memory 52, and devices coupled to a system bus 53. The host processor 50 may include logic circuitry as well as an amount of non-volatile memory 65 used to contain key information. System memory 52 may include but is not limited to conventional memory such as various types of random access memory (RAM) and, for example, DRAM, VRAM, SRAM, etc. as well as memory-mapped I/O devices.

System bus 53 may be implemented in compliance with any type of bus architecture including peripheral component interconnect (PCI) and universal serial bus (USB) and the like.

One of the devices that may be coupled to the system bus 53 is a non-volatile memory 62 which interfaces with the system bus 53 via a bus interface 60. Also connected to system bus 53 can be a removable media reading device, such as a CD-

ROM drive 70. Contained within a non-volatile memory 62 are the software instructions 63 used by the computer system during the system power-up (boot) sequence.

Also stored within the non-volatile memory 62 (alternatively stored in non-volatile memory 65) is the basic input/output software program (BIOS) 64. According to an aspect of the present invention, a signature or key 66 can also be stored in the non-volatile memory. Alternatively the signature can be stored in the non-volatile memory of the host processor itself on another memory location within the computer system. The signature 66 may be stored in a non-volatile memory location that is accessible by the manufacturer during the final assembly process of the computer system. The signature can be an identifier which identifies a computer system.

Referring to Figure 2, Figure 2 is a flowchart which represents steps carried out during the system bootup according to aspects of the present invention as implemented on a FAT (file allocation table) based operating system. These steps can be carried out by the host processor 50 shown in Figure 1 under the control of memory-resident programs, such as software drivers, which reside in the system memory after being loaded.

As represented by block 202, a software driver for the removable media reader which has a CD drive is loaded into the system memory. A driver is software that enables the operating system to communicate with hardware. Typically the driver is loaded from media such as removable media such as a floppy disk or from fixed media such as a hard disk drive. Next, as represented by block 204, a BIOS lock driver is loaded into the system memory. As represented by the block 206, the BIOS lock driver then causes the system to search for the signature, which can be stored, for example, in the non-volatile memory 62 shown in Figure 1. As represented by block 208, if the signature is located, the system, as represented by block 210, sets (unlocks) the decode ability of the BIOS lock driver. However, if the signature is not located, the decode ability is not set, thus remaining locked. As represented by block 212, the BIOS system driver is then loaded into the system memory and includes the name of the BIOS lock driver. When software (for example, the file system driver 220 depicted in Figure 2a and described further below) seeks to or requests to read information from the CD drive,

instead of sending that request to the CD driver, that request will be sent to the BIOS lock driver.

Figure 2a is a block diagram representation of the three drivers discussed in Figure 2. As represented by Figure 2a, the process of Figure 2 installed the file system driver 220 and linked it to the BIOS lock driver 222. The BIOS lock driver is linked to the CD driver 224. Therefore, read requests from the file system driver to the CD drive driver 224 are past through the BIOS lock driver 222.

Referring to Figure 3, Figure 3 is a flowchart representing the process or method which occurs when a removable media, such as a compact or CD, is inserted into the removable media reader according to aspects of the present invention.

As represented by block 302, after a CD is inserted into the CD drive (for example CD drive 70 of Figure 1), the file system driver requests the status of the CD from the BIOS lock driver. As is represented by block 304, the BIOS lock driver passes that request to the CD drive driver. The CD drive can generate a hardware interrupt to flag or notify the computer system that the status of the CD needs to be checked before accessing the CD. The interrupt can be generated because new media has been inserted into the CD drive and could be caused by a number of error conditions.

As represented by block 306, if this is a new CD inserted into the drive, i.e., a CD the status of which has not been already determined, and, as represented by block 308, if the decode ability has been set (see Figure 2), then, as represented by block 310, the BIOS lock driver requests the first sector of the CD from the CD driver. As represented by block 312, the BIOS lock driver determines whether the first sector from the CD is encoded. The decoding process or algorithm of the BIOS lock driver is then set On (enabled) if the sector is encoded or it is set Off (disabled) if the sector is not encoded. As represented by block 314, the BIOS lock driver then passes the insert status of the CD to the file system driver.

Referring again to block 308, if the decode ability of the BIOS lock driver has not been set (see Figure 2), the process proceeds from block 308 directly to block 314. Additionally, referring back to block 306, if the CD is not a new CD (i.e., its status has already been determined), the process proceeds directly from block 306 to block 314.

Referring to Figure 4, Figure 4 is a flowchart which represents the process associated with the computer system reading the information stored on removable media such as a CD. In order for the computer system 10 to retrieve or read information from the removable media in the removable media reader, such as CD drive 70 of Figure 1, as represented by block 402, the file system driver requests sectors on the CD from the BIOS lock driver. As represented by block 404, the BIOS lock driver passes the request to the CD drive driver.

As represented by block 406, the BIOS lock driver determines whether the decoding has been set (see block 308 of Figure 3). If the decoding is set, as represented by block 408, the BIOS lock driver decodes each sector as it is received from the CD drive driver. The sectors are then passed to the file system driver as represented by block 410.

Referring again to block 406, if the decoding is not set, decoding is not performed and the requested sectors are passed to the file system driving without decoding.

We now turn to computer systems, such as system 10 depicted in Figure 1, operating under control of Windows NT/2000 or similar type operating system (referred to as an NT system). In an NT system, there are generally two ways of interfacing with hardware. The first is referred to as monolithic, which means there is just one driver that handles all of the interaction with a piece of hardware. The second is referred to as modular, which means there can be several levels of drivers which are gone through to the hardware. Taking the CD-ROM as a modular example, and referring to Figure 5a, the first driver that gets a request for data from the CD is the CD-ROM file system driver 530. This driver knows the format of data files stored on a CD, as opposed to another file system driver that knows the format of audio stored on a CD. The next level can be one or more upper filter drivers 528, that add value to the data coming from the CD-ROM class function driver. The class function driver 526 defines the base functions that are defined for all CD-ROMs such as read sector, play, stop, rewind, check for data corruption etc. The next level can be one or more lower filter drivers 524, that add value to the data coming from the port drivers 520. The port drivers 520 know how to communicate with the different buses such as SCSI, IDE, etc. Attached to the port

drivers 520 are mini-port drivers 522 that are designed to handle special functions of a device that the port driver does not implement.

Referring to Figure 5, Figure 5 is a flowchart which represents the process or method executed upon bootup of a computer system which is running on an NTFS (NT file system) based operating system according to aspects of the present invention. The process is carried out by the computer system operating under the control of software or firmware, such as a bootup module, as is commonly known in the art.

As represented by block 502, low-level drivers and the class function for driver for CD-ROMS are loaded into the system memory. For example, they can be loaded from removable media or from the hard disk drive.

As represented by block 504, the BIOS lock is then similarly loaded into the system memory. The BIOS lock can be implemented as an upper filter driver (see, Figure 5a, block 528). As represented by block 506, functions are then all set to pass through to the class function driver. Because there are several function that will not be trapped, initially, all functions are set to pass through the BIOS lock.

As represented by block 508, the system then scans for the signature, such as the signature 66 shown in Figure 1. This can be accomplished, for example, by reading the contents of a predetermined non-volatile memory address and comparing those contents with information stored on the removable media. As represented by block 510, if the signature is located the process then proceeds to block 512. In block 512, traps on read and device-control functions are then set so that those functions will pass through the BIOS lock upper filter driver. In other words, when those functions are to be used, control is first passed to the BIOS lock upper filter driver. However, if, as represented by block 510, the signature is not found, the traps on the read and control functions are left (as they were set in block 506) to pass through the BIOS lock upper filter driver without modification.

Referring to Figure 6, Figure 6 is a flowchart which represents the process or method which can be carried out by the BIOS lock upper filter driver to trap the sub function "check verify". This process determines whether a CD is encoded the first time the CD is accessed. The sub function check verify determines, for example, for a

CD drive, how many times a CD has been changed since the driver has been started. That information is stored in a memory location referred to as "change count."

As represented by block 602, if the subfunction is verified, as represented by block 604, a change count is read from the class function driver. As represented by block 606, if the count has changed (indicating, for example, in the case o f a CD drive, that a new CD has been inserted into the drive), as represented by block 608, the first sector is read from the low-level driver. The BIOS lock upper filter driver then determines whether that first sector is encoded. If that sector is encoded, the BIOS lock upper filter driver sets the decoding to On, or if it is not encoded the decoding is not set to On as represented by block 610.

However, referring back to block 606, if the count has not changed, (indicating, for example, in the case of a CD drive, that a new CD has not been inserted) the process proceeds directly to block 612. Similarly, referring back to block 602, if the device control function that has been requested is not check verify, the process proceeds directly to block 612.

As represented by block 612, device-control function is then passed to the class function driver. The process does not break the flow of data between drivers. It simply turns on or off the decode for the read function. Therefore, device control function is passed on to the class function driver. Next, as represented by block 614, the results are then passed to the calling driver.

Referring now to Figure 7, Figure 7 is a flowchart which represents the process or method carried out in a computer operating system, for example, with an NT system, to perform a read function from a removable media device such as a CD drive. As represented by block 702, the read function or read request is passed to the class function driver from a calling driver. As represented by block 704, the BIOS lock upper filter driver determines whether decoding is set On. If the decoding is set On, as represented by block 706, the BIOS lock upper filter driver decodes sectors as they are received from the class function driver. As represented by block 708, those sectors are then passed to the calling driver. However, referring again to block 704, if the decoding is not set On, sectors are passed to the calling driver without being decoded.

Figures 8 and 9 represent processes or methods associated with an aspect of the present invention when the functionality of the detection of encoded data on the removable media, such as represented by blocks 608 and 610 of Figure 6, and the decoding of such data, such as is represented by blocks 704 and 706 of Figure 7, are placed within the firmware of the removable media reader such as the BIOS 71 of the CD drive 70 of Figure 1. The functions represented by Figures 8 and 9 can be implemented by the BIOS software or firmware running on the removable media reader. In such a system the BIOS lock upper filter driver would not be needed.

Referring to Figure 8, as represented by block 802, the CD drive (or other such device) first reads the first sector of data. As represented by block 804, the CD drive sets decoding On or Off based upon its determination of whether the first sector is encoded.

Figure 9 is a flowchart representing aspects of the read function carried out in the firmware of the CD drive. As represented by block 902, the CD drive first reads the sectors requested by the driver. As represented by block 904, if the decoding is set On, the process then proceeds to block 906. As represented by block 906, the sectors are decoded. As represented by block 908, the sectors are then passed to the CD drive driver (224 of Figure 2A). Referring back to block 1004, if the decoding is not set On, the process proceeds directly to block 908.

In a system operating as represented by Figures 8 and 9, the process depicted in Figure 8 would take place each time a new CD was inserted into the drive. The CD drive BIOS recognizes when a new CD is inserted. Because the BIOS will only recognize a specific encoding scheme, no signature is needed. (However, a signature could be used to provide additional security.) In other words, a CD that is encoded with a different scheme would not be recognized as encoded by the process of Figure 8. Therefore, decode would not be set ON. When data was read from the CD in accordance with the process of Figure 9, it would not be decoded and would be sent in an unrecognizable form to the computer.

While this invention has been described with reference to specific embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the described embodiments, as well as other embodiments of

the invention which are apparent to persons skilled in the art, are deemed to lie within the spirit and scope of the invention. For example, though embodiments of the invention have been described with reference to a personal computer, the invention can be implemented in other types of computer systems. Additionally, though the invention has been described with reference to DOS and Windows NT operating systems, it will be apparent to those of ordinary skill in the art that this system can also be readily adapted to other operating systems.

WHAT IS CLAIMED IS:

1. A computer system for controlling the unauthorized use of software, comprising:

a host processor;

a removable media reading device coupled to said host processor;

a non-volatile memory coupled to said host processor, said non-volatile memory including a predetermined location for storing a signature; and

a bootup program stored in said non-volatile memory, said bootup program configured such that upon execution by said host processor the computer system will only be configured to decode encoded media in said removable media reading device if said signature is located in said predetermined location.

2. A method of reading encoded data from a removable media device in a computer system, the method comprising:

initiating a request for data from the removable media reader;

determining whether the system is authorized to decode encoded data;

if the system is authorized to decode encoded data, determining whether the first sector on the removable media is encoded; and

if the system is authorized to decode encoded data and the first sector on the removable media is encoded, decoding the requested data on the removable media.

3. The method of Claim 2, wherein determining whether the system is authorized to decode encoded data includes verifying that a signature is stored in the system in a predetermined memory location.

4. The method of Claim 2, further comprising if the decode ability is not set, passing data, whether encoded or not, to the system.

5. The method of Claim 2 further comprising if the decode ability is set and the decoding is set OFF, passing data to the system from the removable media as it is requested.

6. A method of preventing unauthorized access to encoded content stored on removable media by a computer system running an operating system and having a removable media reader, the method comprising:

running a memory-resident program with a lockable decoding function to control the interaction between the operating system and the removable media reader;

scanning the computer system for a predetermined signature and unlocking said lockable decoding function if said signature is found;

examining a removable medium to determine whether it is encoded and enabling said lockable decoding function if the removable medium is encoded; and

decoding data from the removable medium if said lockable decoding function is enabled.

7. The method of Claim 6, wherein said memory-resident program is inserted between the operating system and a device driver for the removable media reader.

8. The method of Claim 6, wherein said memory-resident program adds at least one modular driver between the operating system and the removable media reader.

9. The method of Claim 6, wherein said predetermined signature is stored in a basic input output system (BIOS) circuitry of said computer system.

10. The method of Claim 6, wherein said predetermined signature is stored in a non-volatile memory of a central processing unit of said computer system.

11. A method of preventing unauthorized access to encoded content stored on removable media by a computer system running an operating system and having a removable media reader, the method comprising:

inserting a driver with a lockable decoding function between the operating system and the device driver for the removable media reader to control the transfer of information between the operating system and the removable media reader;

scanning the computer system for a predetermined signature, and unlocking said lockable decoding function if said signature is found;

examining a removable medium to determine whether it is encoded and enabling said lockable decoding function if the removable medium is encoded; and

decoding data from the removable medium if said lockable decoding function is enabled.

12. The method of Claim 11, wherein said predetermined signature is stored in a basic input / output system (BIOS) circuitry of said computer system.

13. The method of Claim 11, wherein said predetermined signature is stored in a non-volatile memory of a central processing unit of said computer system.

14. The method of Claim 11, wherein said examination of a removable medium to determine whether it is encoded further comprises:

trapping insert status requests of the removable media reader from the operating system to the device driver of said removable media reader.

15. The method of Claim 11, wherein said decoding of data from the removable medium further comprises:

trapping read requests from the operating system to the device driver of said removable media reader.

16. A method of preventing unauthorized access to encoded content stored on removable media by a computer system running an operating system and having a removable media reader, the method comprising:

adding at least one modular driver between the operating system and the removable media reader to incorporate a lockable decoding function to control the transfer of information between the operating system and the removable media reader;

scanning the computer system for a predetermined signature and unlocking said lockable decoding function if said signature is found;

examining a removable medium to determine whether it is encoded and enabling said lockable decoding function if the removable medium is encoded; and

decoding data from the removable medium if said lockable decoding function is enabled.

17. The method of Claim 16, wherein said predetermined signature is stored in a basic input / output system (BIOS) circuitry of said computer system.

18. The method of Claim 16, wherein said predetermined signature is stored in a non-volatile memory of a central processing unit of said computer system.

19. The method of Claim 16, wherein the lockable decoding function is incorporated by adding an upper filter modular driver with said lockable decoding function.

20. The method of Claim 16, wherein said examination of a removable medium to determine whether it is encoded further comprises:

trapping insert status requests of the removable media reader from the operating system to the class function driver of said removable media reader.

21. The method of Claim 16, wherein said decoding of data from the removable medium further comprises:

trapping read requests from the operating system to the class function driver of said removable media reader

22. A method of preventing unauthorized access to encoded content stored on removable media by a computer system running an operating system and having a removable media reader, the method comprising:

programming the firmware of the removable media reader to incorporate a lockable decoding function to control the transfer of information between the operating system and the removable media reader;

scanning the computer system for a predetermined signature, and unlocking said lockable decoding function if said signature is found;

examining a removable medium in the removable media reader to determine whether it is encoded, and enabling said lockable decoding function if the removable medium is encoded; and

decoding data from the removable medium if said lockable decoding function is enabled.

23. The method of Claim 22, wherein said predetermined signature is stored in a basic input / output system (BIOS) circuitry of said computer system.

24. The method of Claim 22, wherein said predetermined signature is stored in a non-volatile memory of a central processing unit of said computer system.

25. The method of Claim 22, wherein said unlocking and said enabling of said lockable decoding function is performed by a memory-resident program.

26. A computer system for preventing unauthorized access to encoded content stored on removable media, comprising:

a removable media reader; and

a host processor running an operating system, said host processor coupled to said removable media reader and configured to

control the transfer of information between the operating system and the removable media reader with a lockable decoding function,

scan for a predetermined signature and unlock said lockable decoding function if said signature is found,

examine a removable medium in said removable media reader to determine whether it is encoded and enable said lockable decoding function if the removable medium is encoded, and

decode data from the removable medium if said lockable decoding function is enabled.

BIOS LOCK ENCODE/DECODE DRIVER

Abstract of the Disclosure

Systems and methods for preventing the unauthorized access to data stored on removable media, such as software, include storing a predetermined signature in the area of non-volatile memory in a computer system. Upon initialization of the computer system, a check is made to verify the signature. Only if the signature is verified will decoding software operate.

S:\DOCS\REC\REC-3913.DOC
061500

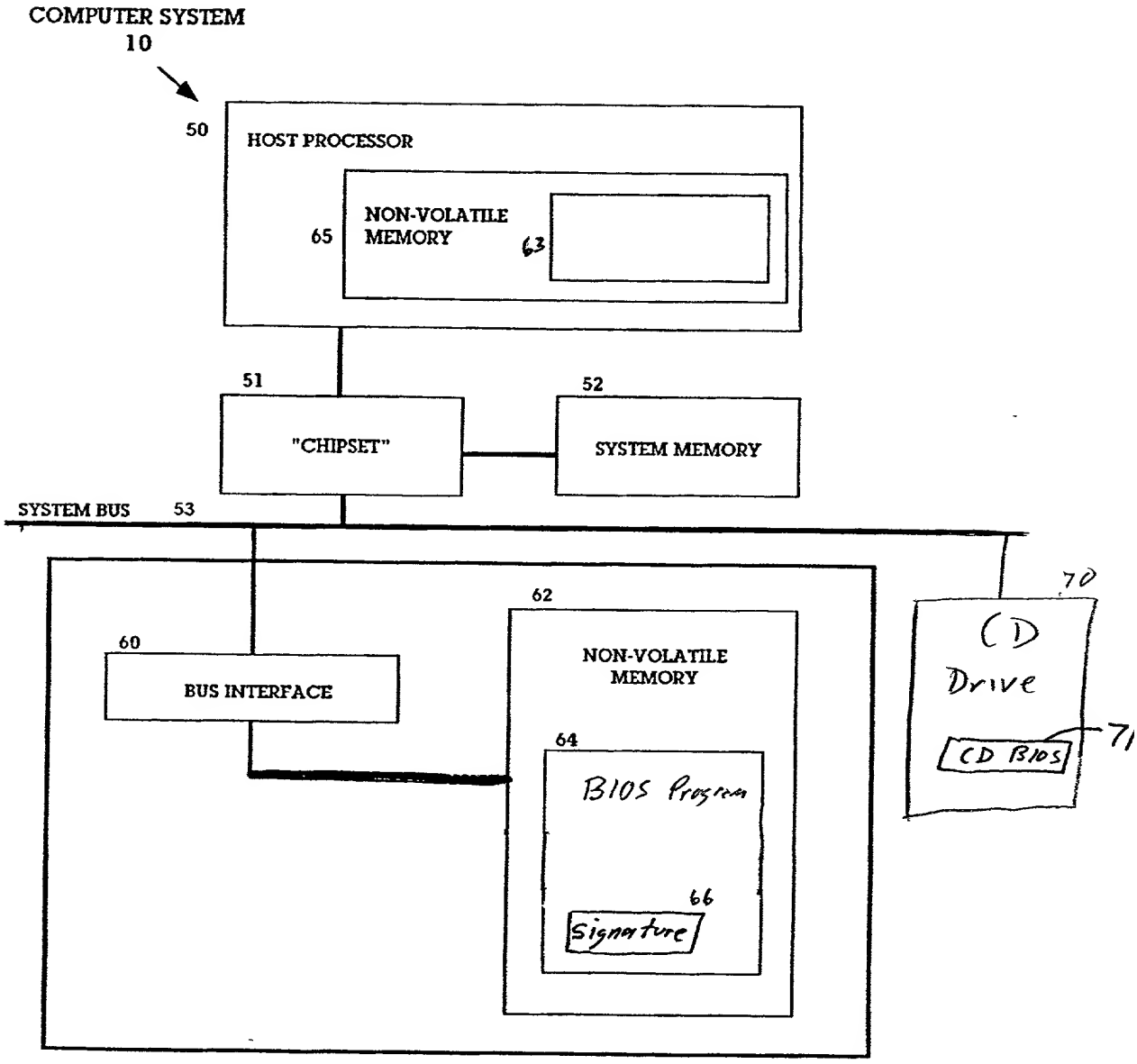

COMPUTER SYSTEM
10
50
HOST PROCESSOR
65
NON-VOLATILE MEMORY
63
51
"CHIPSET"
52
SYSTEM MEMORY
SYSTEM BUS
53
60
BUS INTERFACE
62
NON-VOLATILE MEMORY
64
BIOS Program
66
Signature
70
CD Drive
CD BIOS
71

FIGURE 1

Fig. 2

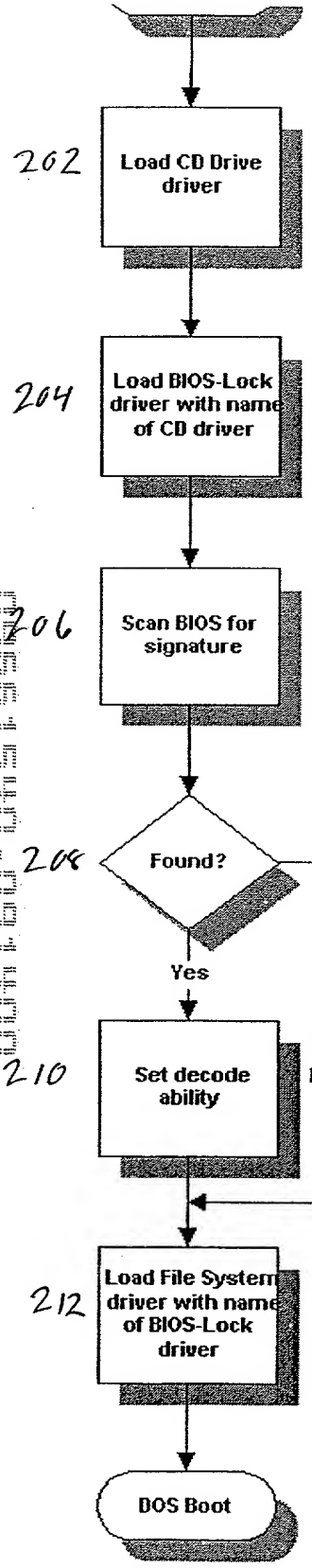


Fig. 3

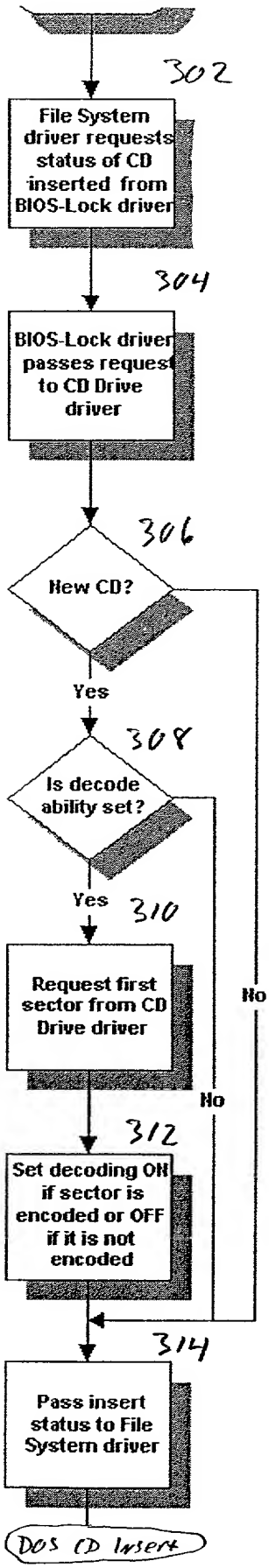


Fig. 4

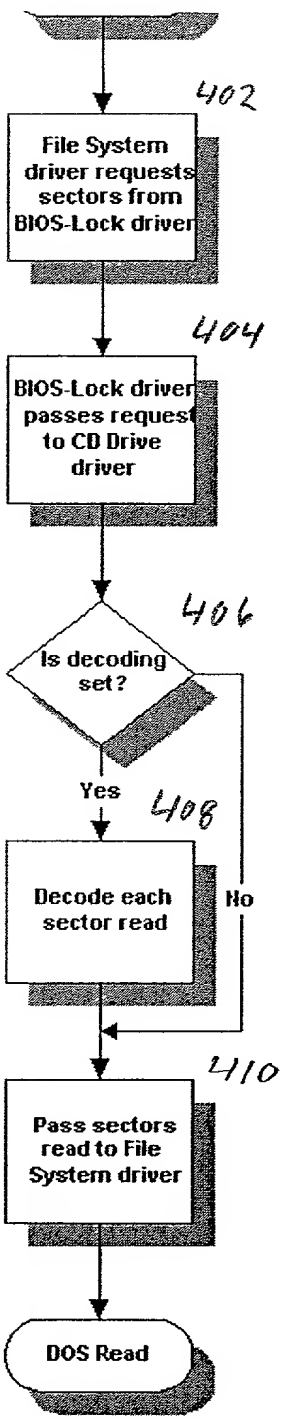


Fig 2a

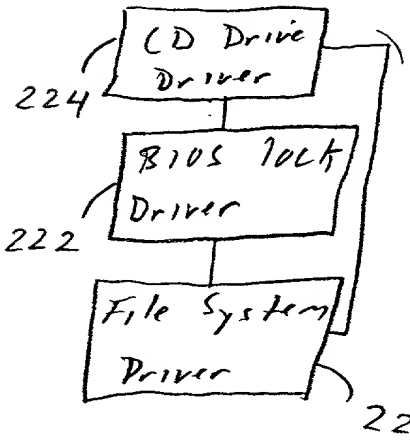


Fig. 5

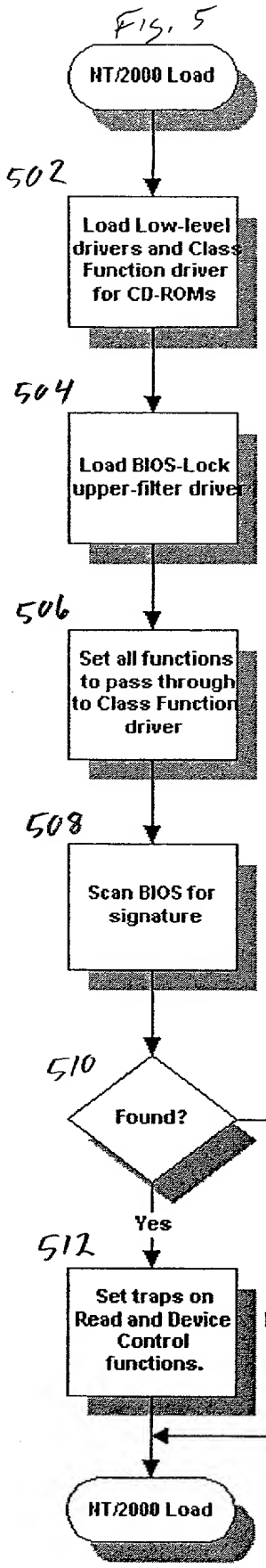


Fig. 6

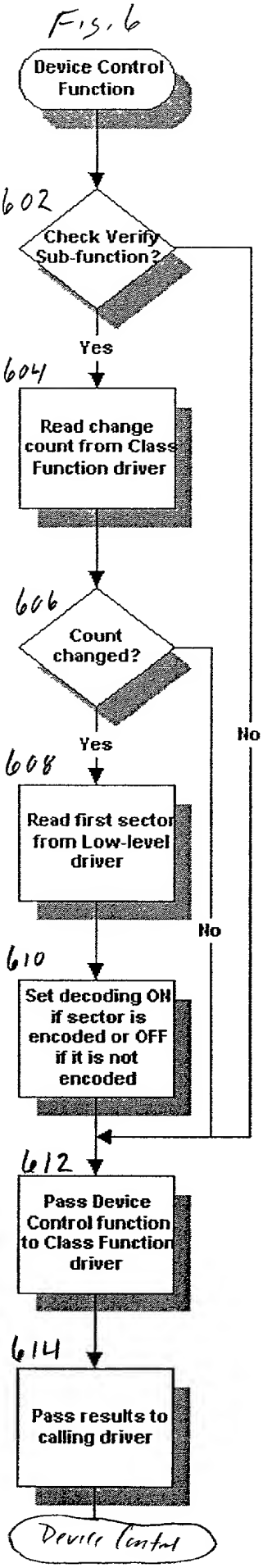
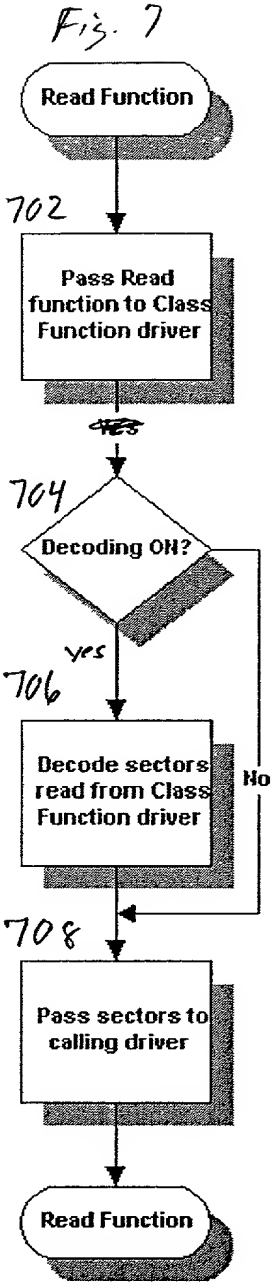


Fig. 7



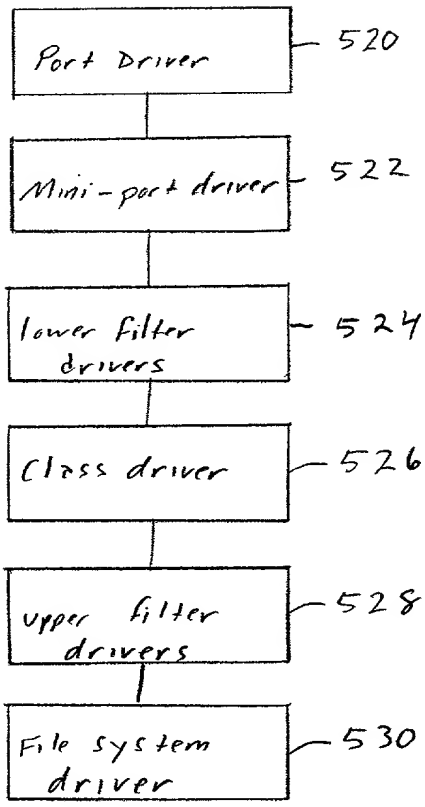


Fig. 8

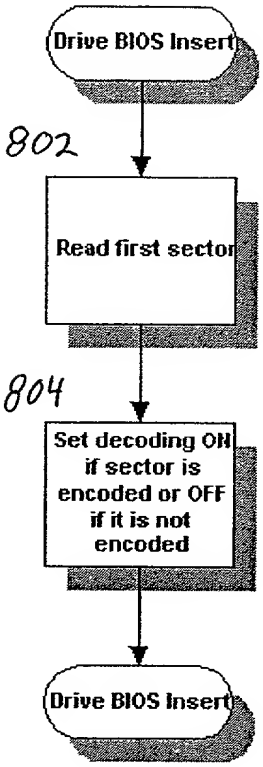
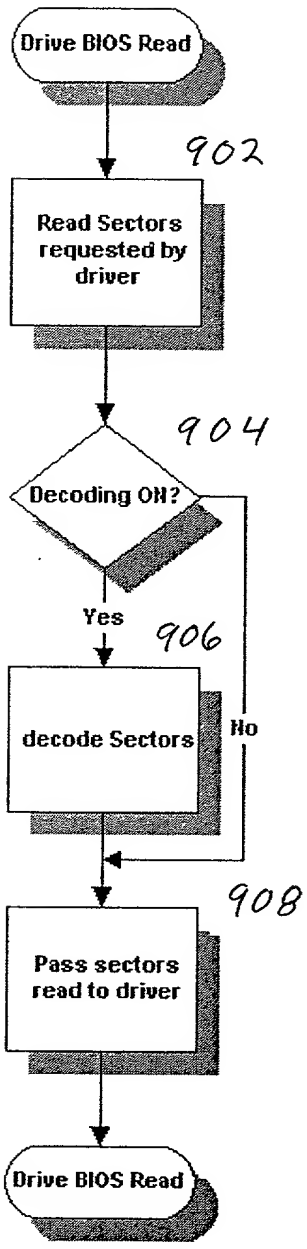

Drive BIOS Insert
802
Read first sector
804
Set decoding ON if sector is encoded or OFF if it is not encoded
Drive BIOS Insert

Fig. 9


Drive BIOS Read
902
Read Sectors requested by driver
904
Decoding ON?
Yes
906
decode Sectors
No
908
Pass sectors read to driver
Drive BIOS Read

DECLARATION - USA PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled **BIOS LOCK CD-ROM ENCODE/DECODE DRIVER**; the specification of which is attached hereto;

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above;

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56;

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful, false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole inventor: **Duane Allen**

Inventor's signature _____

Date 8.15.00

Residence: **232 Buckskin Drive, Nampa, Idaho 83687**

Citizenship: **U.S.**

Post Office Address:

Send Correspondence To:
KNOBBE, MARTENS, OLSON & BEAR, LLP
Customer No. 20,995

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	MICRON ELECTRONICS, INC.	)
App. No.	:	Unknown	)
Filed	:	Herewith	)
For	:	BIOS LOCK CD-ROM ENCODE/DECODE DRIVER	)
Examiner	:	Not Assigned	)

ESTABLISHMENT OF RIGHT OF ASSIGNEE TO TAKE ACTION
AND
REVOCATION AND POWER OF ATTORNEY

Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

The undersigned is empowered to act on behalf of the assignee below (the "Assignee"). A true copy of the original Assignment of the above-captioned application from the inventor(s) to the Assignee is attached hereto. This Assignment represents the entire chain of title of this invention from the Inventor(s) to the Assignee.

I declare that all statements made herein are true, and that all statements made upon information and belief are believed to be true, and further, that these statements were made with the knowledge that willful, false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001, and that willful, false statements may jeopardize the validity of the application, or any patent issuing thereon.

The undersigned hereby revokes any previous powers of attorney in the subject application, and hereby appoints the registrants of Knobbe, Martens, Olson & Bear, LLP, 620 Newport Center Drive, Sixteenth Floor, Newport Beach, California 92660, Telephone (949) 760-0404, **Customer No. 20,995**; also Steven P. Arnold, Registration No. 33,354, Hoyt A. Fleming, III, Registration No. 41,752, and Paul A. Revis, Registration No. 45,040, of Micron Electronics, Inc., 900 East Karcher Road, Nampa, Idaho 83687, Telephone (208) 898-3434, as its attorneys with full power of substitution and revocation to prosecute this application and to transact all business in the U.S. Patent and Trademark Office connected herewith. This

App. No. : Unknown
Filed : Herewith

appointment is to be to the exclusion of the inventor(s) and his attorney(s) in accordance with the provisions of 37 C.F.R. § 3.71.

Please use **Customer No. 20,995** for all communications.

MICRON ELECTRONICS, INC.

Dated: August 15, 2000

By: _____
Paul A. Revis

Title: Intellectual Property Counsel

Address: 900 East Karcher Road
Nampa, Idaho 83687

S:\DOCS\REC\REC-4119.DOC
072600